Investigating positional information in the Transformer

Group 9

Outline

- **Background & Motivation**
- Related Work
  - Towards Understanding Position Embeddings
  - Do We Need Word Order Information for Cross-Lingual Sequence Labeling
  - Revealing the Dark Secrets of BERT
  - Accessing the Ability of Self-Attention Networks to Learn Word Order
- Probing for Position: Diagnostic Classifiers (DC) and Perturbed Training
  - Research Questions
  - Experiments & Tasks
- Initial results: DC on BERT, finetuning without positional embeddings

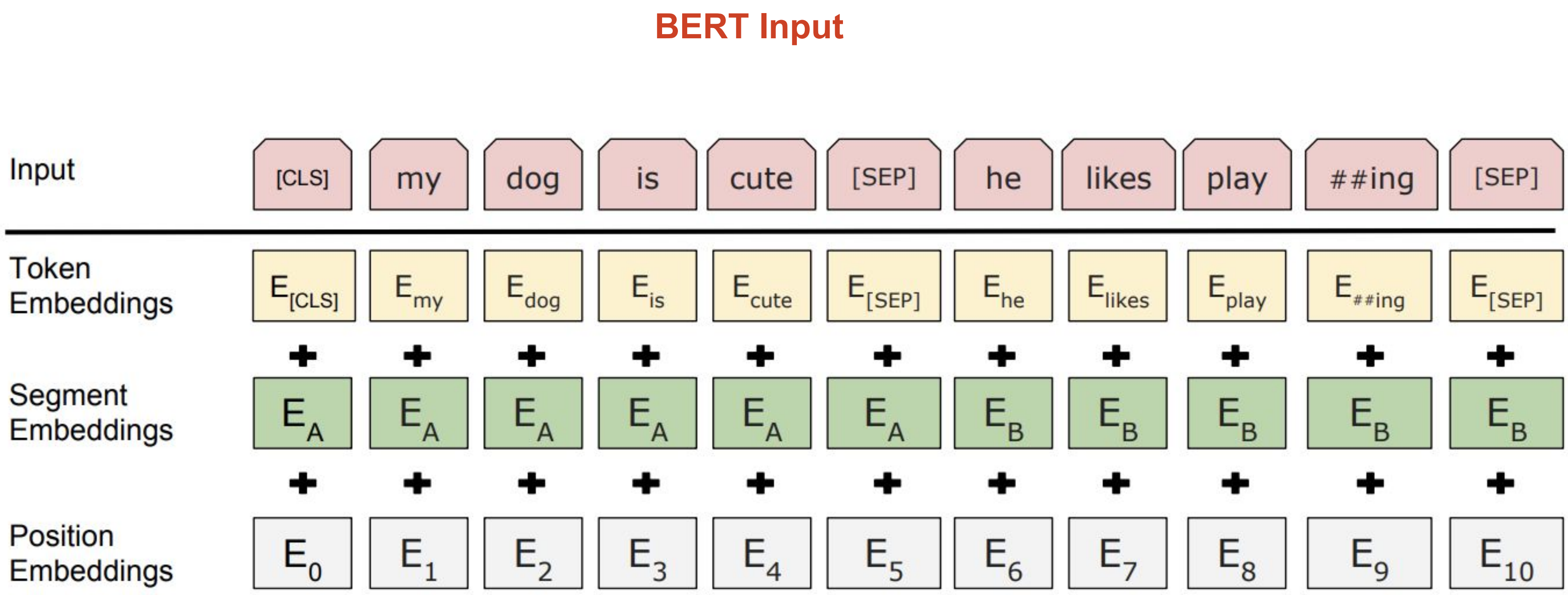
Background & Motivations

- Emergence of self-attention based models (e.g. Transformer, BERT) due to expensive sequential computation (e.g. RNN)
- Adding positional embeddings are the only ways to compensate the word order information captured in sequential models

$$f(j, pos) = f_{we}(j) + f_{pe}(pos)$$

- Positional embeddings/encodings have been comparatively understudied compared with word/sentence embeddings
- Absolute and Relative

Absolute Positional Embeddings



Outline

- Background & Motivation
- **Related Work**
  - Towards Understanding Position Embeddings
  - Do We Need Word Order Information for Cross-Lingual Sequence Labeling
  - Revealing the Dark Secrets of BERT
  - Accessing the Ability of Self-Attention Networks to Learn Word Order
- Probing for Position: Diagnostic Classifiers (DC) and Perturbed Training
  - Research Questions
  - Experiments & Tasks
- Initial results: DC on BERT, finetuning without positional embeddings

Towards Understanding Position Embeddings

Rishi Bommasani
Department of Computer Science
Cornell University
Ithaca, NY, 14853, USA
rb724@cornell.edu

Claire Cardie
Department of Computer Science
Cornell University
Ithaca, NY, 14853, USA
cardie@cs.cornell.edu

Towards Understanding Position Embeddings

- First work on probing positional embeddings of pretrained transformer based language models (BERT & GPT)
- Poses three questions towards understanding positional embedding
  - How are position embeddings produced by different models related?
  - How should we encode position?
  - Are position embeddings transferrable?
- Provides introductory results in tackling the first question

Whether Positional Embeddings are Comparable?

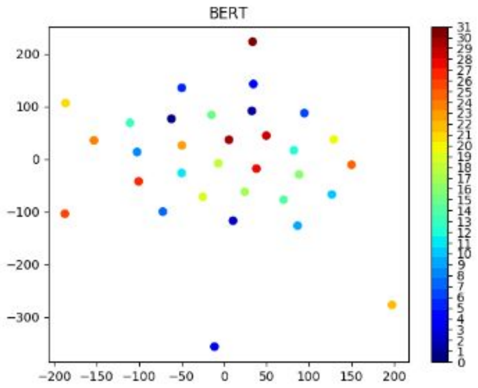
- Tokenization
    - BERT’s Tokenizer WordPiece for English)
    - GPT’s Tokenizer (BPE)
    - A simple white space tokenization algorithm which we found closely modeled our naïve judgments about absolute position

	Reference		
	Human	BERT	GPT
Human	100	21.19	18.84
BERT	25.42	100	48.05
GPT	22.78	48.46	100

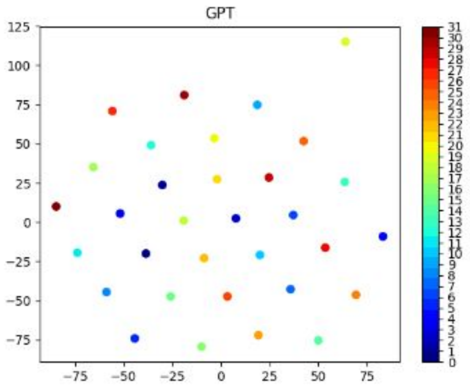
Table 1: Average token alignment is given by the percentage of tokens in the reference that match the token at the same position in the candidate.

Comparison Between BERT & GPT

- Geometry
  - Tightness of clustering
  - Nearest neighbor sets



(a) BERT



(b) GPT

Do We Need Word Order Information for Cross-lingual Sequence Labeling

Zihan Liu, Pascale Fung
Center for Artificial Intelligence Research (CAiRE)
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
zliucr@connect.ust.hk, pascale@ece.ust.hk

Positional Embeddings for Cross-Lingual Tasks

- Hypothesis
  - Cross-lingual models that fit into the source language word order might fail to handle target languages whose word orders are different

- Experiment Setup
  - Zero-shot learning for various tasks (POS, NER, etc)
  - Initialize word/position embeddings from mBERT
  - For all the tasks, use English as the source language and other languages as target languages.
  - Do not use any data sample in target languages, and select the final model based on the performance on the source language dev set

Positional Embeddings for Cross-Lingual Tasks

Accuracy on the POS task

	es	fr	pt	ru	el	AVG
TRS+Linear	72.08	79.03	32.65	78.06	72.75	66.91
OATRS+Linear	72.70	80.16	33.05	**78.64**	75.01	67.91
SHTRS+Linear	72.21	78.43	32.81	77.82	75.48	67.35
SHOATRS+Linear	**72.65**	**80.99**	**35.84**	76.70	**75.69**	**68.37**
mBERT+Linear (Fine-tune mBERT)						
w/ word order	84.31	89.05	54.30	84.19	84.35	79.24
w/o word order	**84.73**	**89.18**	**54.56**	**86.17**	**85.66**	**80.06**

F1 on the NER task

	es	de	nl	AVG
TRS+Linear	57.43	47.78	63.15	56.12
OATRS+Linear	58.29	45.97	66.34	56.87
SHTRS+Linear	59.29	42.99	**67.09**	56.46
SHOATRS+Linear	**61.35**	**46.54**	65.35	**57.75**
mBERT+Linear (fine-tune mBERT)				
w/ word order	67.80	**65.71**	71.95	68.49
w/o word order	**69.33**	65.60	**73.18**	**69.37**

TRS: Transformer (8 heads)
OATRS: Order-agnostic Transformer (8 heads)
SHTRS: Single-head Transformer (1 head)
SHOATRS: Single-head Order-agnostic Transformer (1 head)

Revealing the Dark Secrets of BERT

Olga Kovaleva, Alexey Romanov, Anna Rogers, Anna Rumshisky
Department of Computer Science
University of Massachusetts Lowell
Lowell, MA 01854
{okovalev,arum,aromanov}@cs.uml.edu

Revealing the Dark Secrets of BERT

- Questions investigated:
  - What are the common attention patterns, how do they change during fine-tuning, and how does that impact the performance on a given task?
  - What linguistic knowledge is encoded in self-attention weights of the fine-tuned models and what portion of it comes from the pretrained BERT?
  - How different are the self-attention patterns of different heads, and how important are they for a given task?

Positional Information in Self-Attention Maps

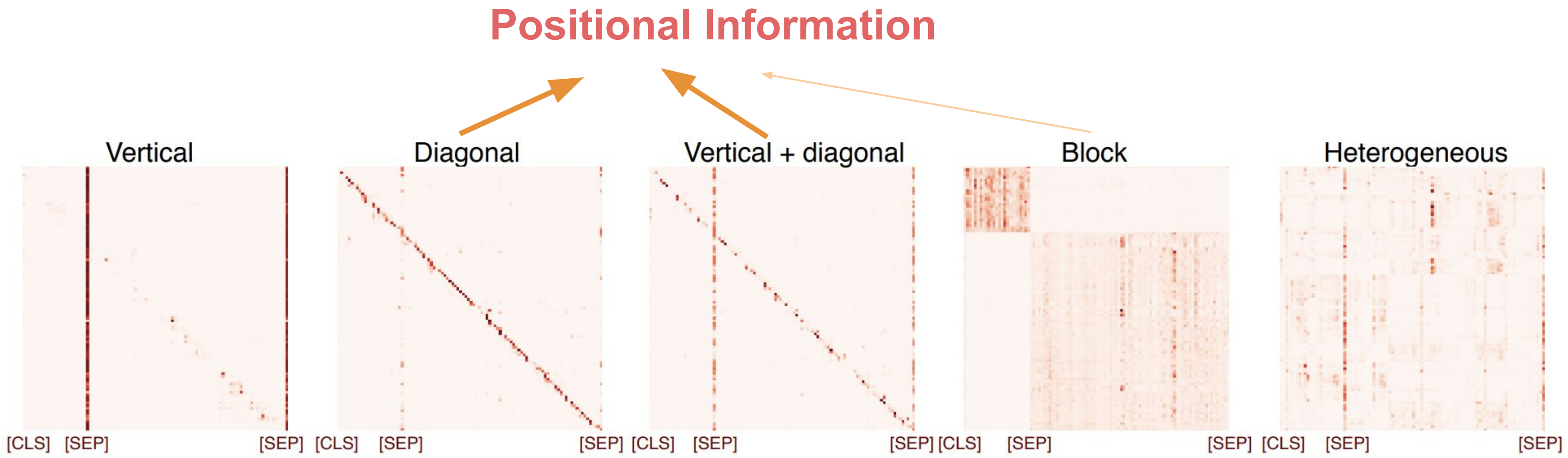


Figure 1: Typical self-attention classes used for training a neural network. Both axes on every image represent BERT tokens of an input example, and colors denote absolute attention weights (darker colors stand for greater weights). The first three types are most likely associated with language model pre-training, while the last two potentially encode semantic and syntactic information.

Self-attention Classes for Downstream Tasks

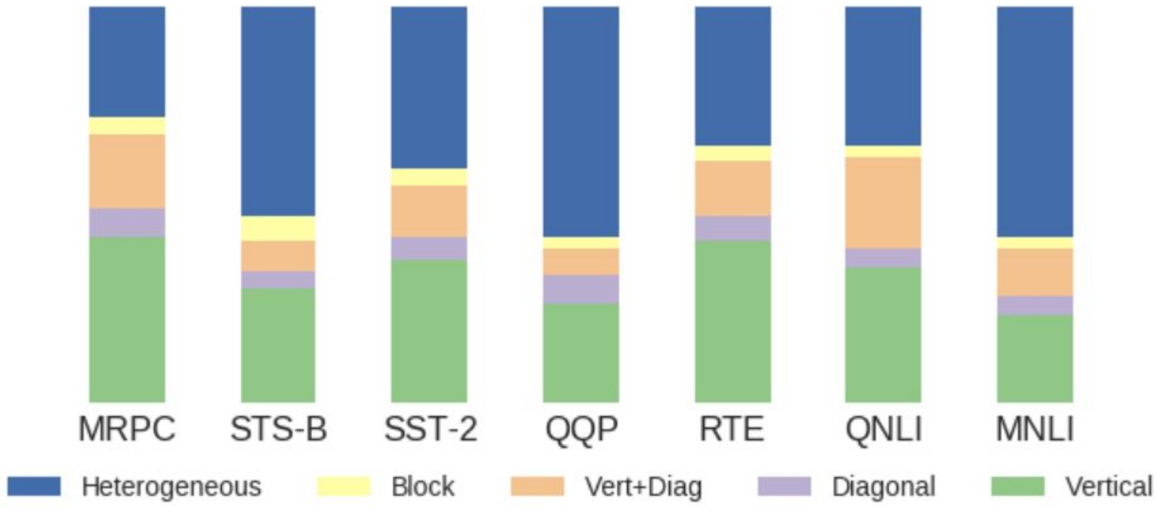


Figure 2: Estimated percentages of the identified self-attention classes for each of the selected GLUE tasks.

Assessing the Ability of Self-Attention Networks to Learn Word Order

Baosong Yang[†] **Longyue Wang**[‡] **Derek F. Wong**[†] **Lidia S. Chao**[†] **Zhaopeng Tu**[‡*]

[†]NLP^{2}CT Lab, Department of Computer and Information Science, University of Macau

nlp2ct.baosong@gmail.com, {derekfw,lidiasc}@umac.mo

[‡]Tencent AI Lab

{vinnylywang,zptu}@tencent.com

Accessing the Ability of Self-Attention Networks to Learn Word Order

- Focus on the following research questions
  - Is recurrence structure obligate for learning word order?
  - Is the model architecture the critical factor for learning word order in the downstream tasks such as machine translation?
  - Is position embedding powerful enough to capture word order information for SAN?

Ability of Self-Attention Networks (SAN) to Learn Word Order

A Probing Task

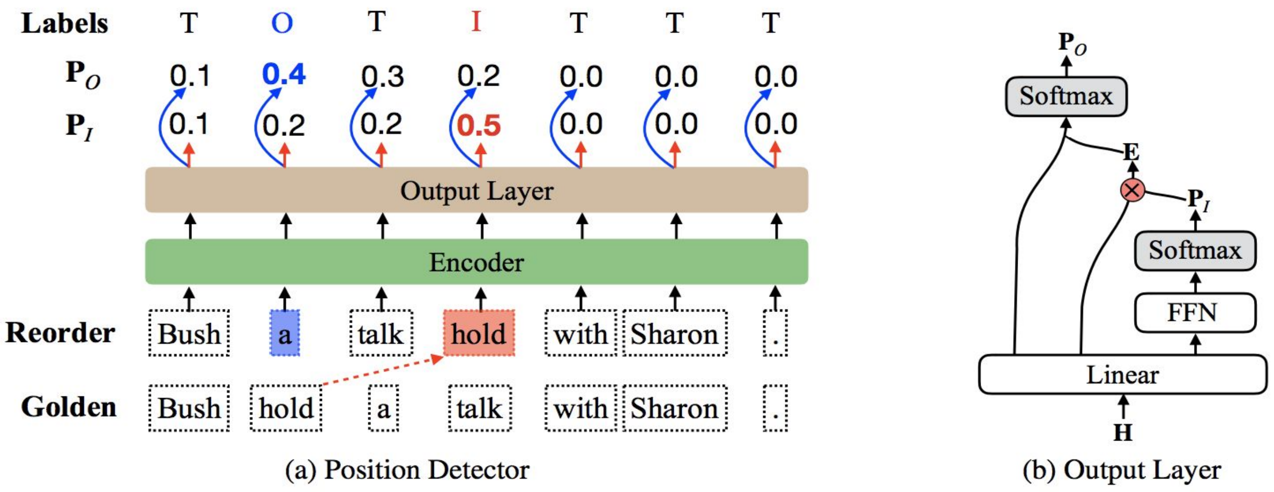


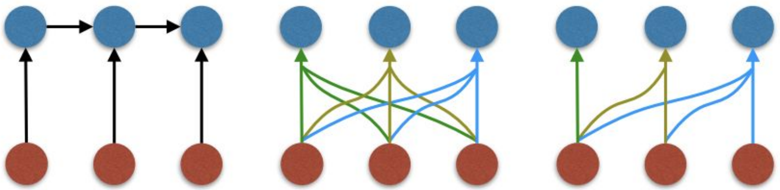
Figure 1: Illustration of (a) the position detector, where (b) the output layer is build upon a randomly initialized or pre-trained encoder. In this example, the word "hold" is moved to another place. The goal of this task is to predict the inserted position "I" and the original position "O" of "hold".

Compare SAN vs RNN

Trained on the word reordering detection (WRD) task data

Models	Insert	Original	Both
RNN	78.4	**73.4**	**68.2**
SAN	73.2	66.0	60.1
DiSAN	**79.6**	70.1	68.0

Table 1: Accuracy on the WRD task. "Insert" and "Original" denotes the accuracies of detecting the inserted and original positions respectively, and "Both" denotes detecting both positions.



(a) RNN (b) SAN (c) DiSAN

Figure 2: Illustration of (a) RNN; (b) SAN; and (c) DiSAN. Colored arrows denote parallel operations.

Compare SAN vs RNN

- **First train (both encoder and decode) on bilingual NMT corpus**
- **Then fix the parameters of the encoder, only train the parameters of the output layer on WRD data**

Model	**Translation**		**Detection**		
	En⇒De	En⇒Ja	En⇒De Enc.	En⇒Ja Enc.	WRD Enc.
RNN	26.8	42.9	33.9	29.0	**68.2**
SAN	27.3	43.6	**41.6**	**32.8**	60.1
- Pos_Emb	11.5	–	0.3	–	0.3
DiSAN	**27.6**	**43.7**	39.7	31.2	68.0
- Pos_Emb	27.0	43.1	40.1	31.0	62.8

Table 2: Performances of NMT encoders pre-trained on WMT14 En⇒De and WAT17 En⇒Ja data. "Translation" denotes translation quality measured in BLEU scores, while "Detection" denotes the accuracies on WRD task. "En⇒De Enc." denotes NMT encoder trained with translation objective on the En⇒De data. We also list the detection accuracies of WRD encoders ("WRD Enc.") for comparison. "- Pos_Emb" indicates removing positional embeddings from SAN- or DiSAN-based encoder. Surprisingly, *SAN-based NMT encoder achieves the best accuracy on the WRD task*, which contrasts with the performances of WRD encoders (the last column).

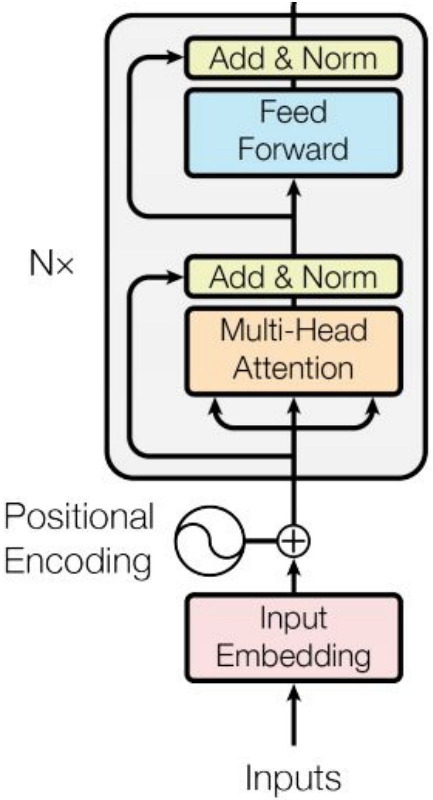
Outline

- Background & Motivation
- Related Work
  - Towards Understanding Position Embeddings
  - Do We Need Word Order Information for Cross-Lingual Sequence Labeling
  - Revealing the Dark Secrets of BERT
  - Accessing the Ability of Self-Attention Networks to Learn Word Order
- **Probing for Position: Diagnostic Classifiers (DC) and Perturbed Training**
  - Research Questions
  - Experiments & Tasks
- Initial results: DC on BERT, finetuning without positional embeddings

Research Questions

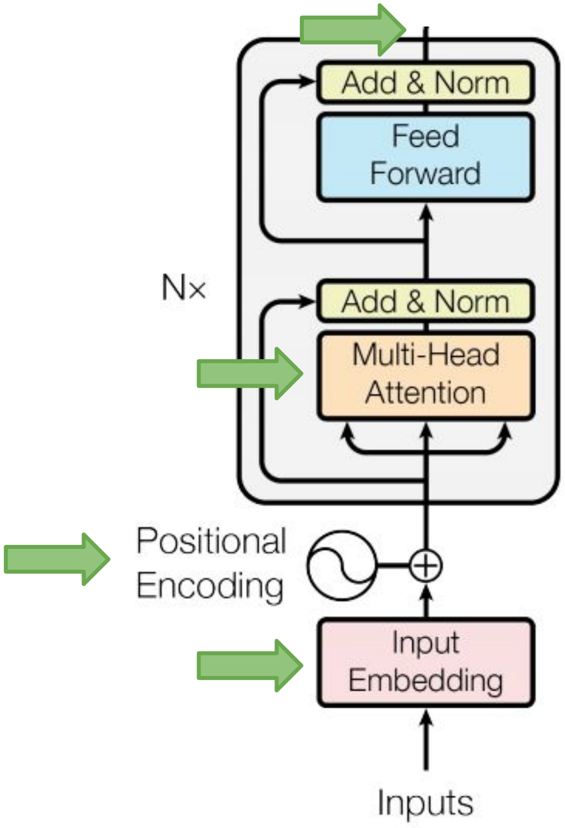
1. What positional information is contained in different parts of the Transformer architecture?
2. How important are positional embeddings (and positional information in general) for different types of NLP tasks?

Position Prediction with Diagnostic Classifiers



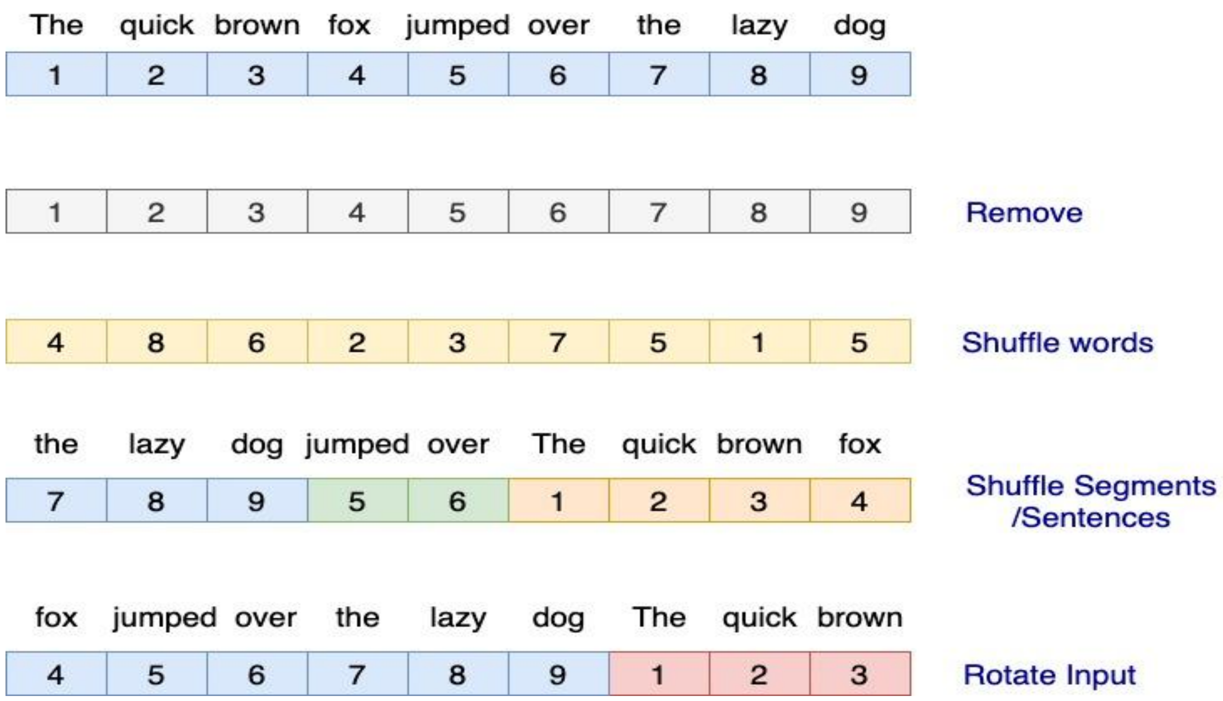
Train a single feed-forward layer to predict the absolute position of each input to BERT at various points in the model

Position Prediction with Diagnostic Classifiers



Train a single feed-forward layer to predict the absolute position of each input to BERT at various points in the model

Perturbed Training for BERT



Experimental Setup and Evaluation

Experiment	Model	Evaluation metrics
Diagnostic Classifier	Pre-trained BERT	DC Metrics
	Fine-tuned BERT	DC Metrics
Perturbation Training	Pre-trained BERT	LM Perplexity
	Fine-tuned BERT	Task Metrics

Table 3: Experimental Setup

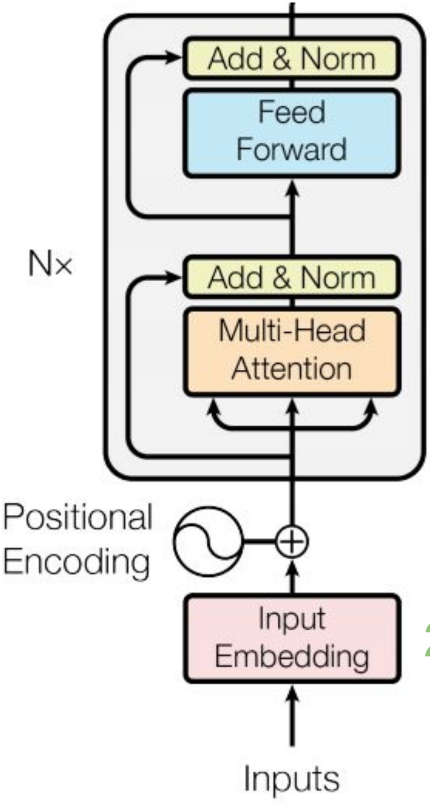
Task	Dataset	Evaluation metrics
Syntax parsing	Universal Dependencies (UD)	LAS/UAS
Coreference resolution	OntoNotes	F1
Summarization	CNN/Daily mail	ROUGE
Text Classification	20 News Group/SST	Accuracy
GLUE NLI Tasks	RTE, MNLI	GLUE NLI metrics
QA Tasks	SQuaD/BoolQ/SWAG	F1/Accuracy

Table 3: Downstream Tasks

Outline

- Background & Motivation
- Related Work
  - Towards Understanding Position Embeddings
  - Do We Need Word Order Information for Cross-Lingual Sequence Labeling
  - Revealing the Dark Secrets of BERT
  - Accessing the Ability of Self-Attention Networks to Learn Word Order
- Probing for Position: Diagnostic Classifiers (DC) and Perturbed Training
  - Research Questions
  - Experiments & Tasks
- **Initial results: DC on BERT, finetuning without positional embeddings**

Position Prediction Accuracy on BERT

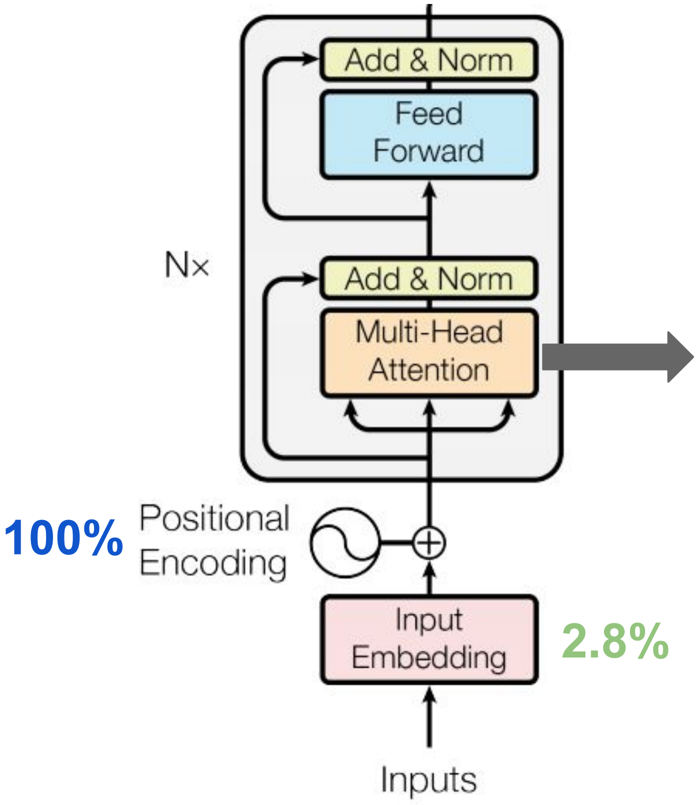


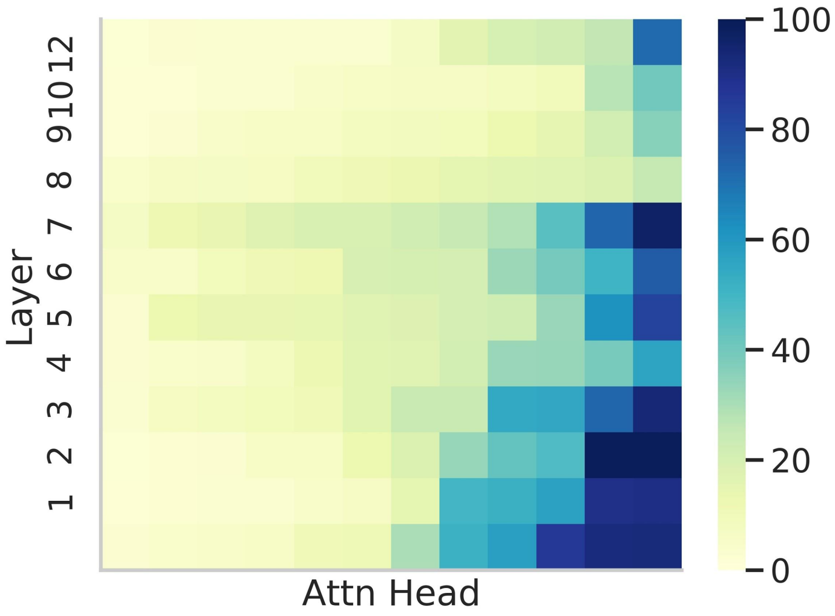
100%

2.8%

Random guessing = 1/512 ≈ 0.2%

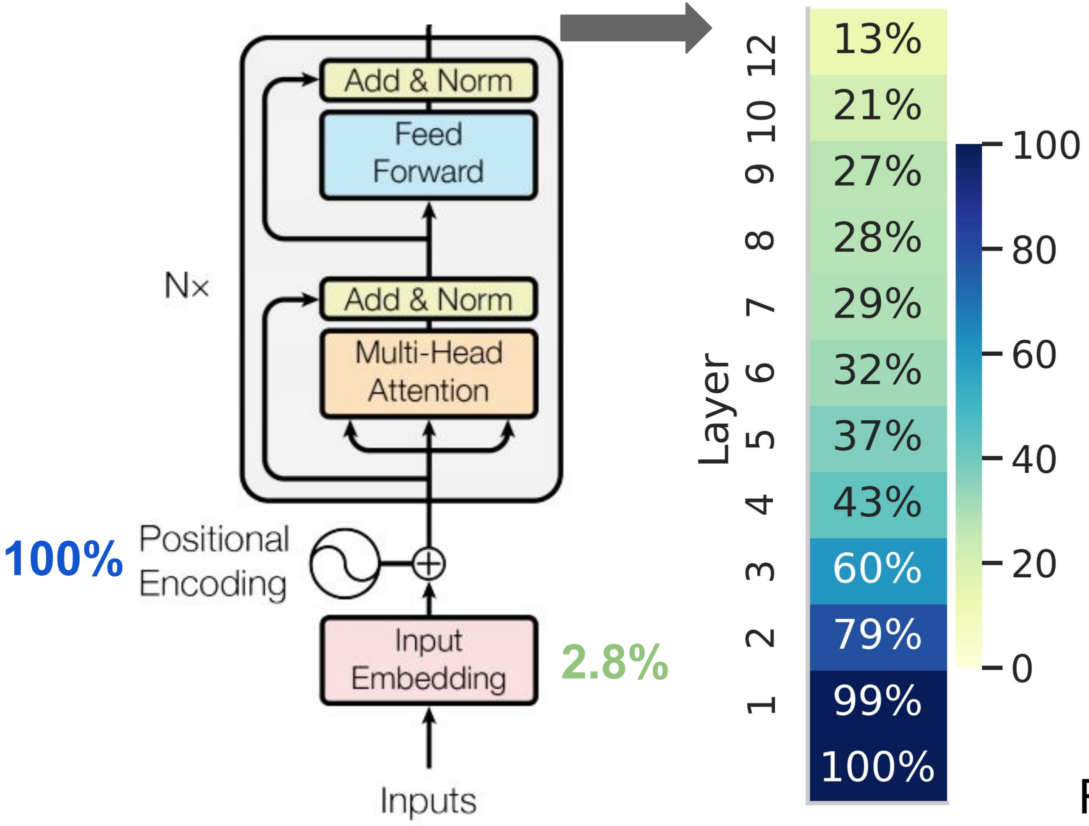
Initial Position Prediction Accuracy on BERT





Random guessing = 1/512 ≈ 0.2%

Initial Position Prediction Accuracy on BERT



Random guessing = 1/512 ≈ 0.2%

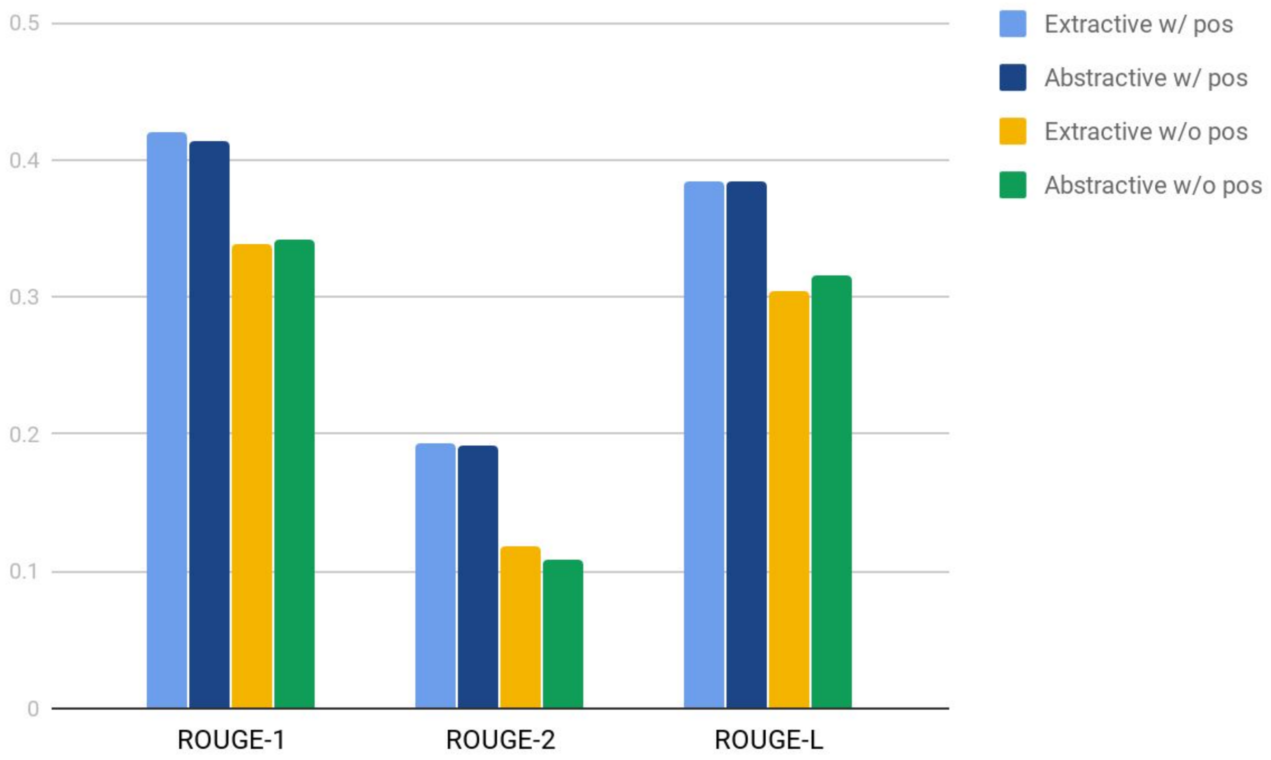
Results on removing position embeddings in BERT

Task Category	Task	with Pos	w/o Pos	Abs Diff	% Diff
Span Extraction	SQuAD (F1)	87.5	29.9	57.6	65.8
Input Tagging	Coreference Resolution (F1)	67.4	44.6	22.8	33.8
Sentence Decoding	CNN/Daily mail (Abstractive summarization)	0.191	0.109	0.08	42.9
Sentence Classification	CNN/Daily mail (Extractive summarization)	0.193	0.119	0.07	38.3
Classification	SWAG (Accuracy)	79.1	66.7	12.4	15.7
Classification	SST (Accuracy)	92.4	87.0	5.4	5.8
Classification	MNLI	80.4	76.9	3.5	4.4
Classification	MNLI-MM	81.0	76.8	4.2	5.2
Classification	RTE	65.0	58.8	6.2	9.5
Classification	QNLI	87.5	83.6	3.9	4.5

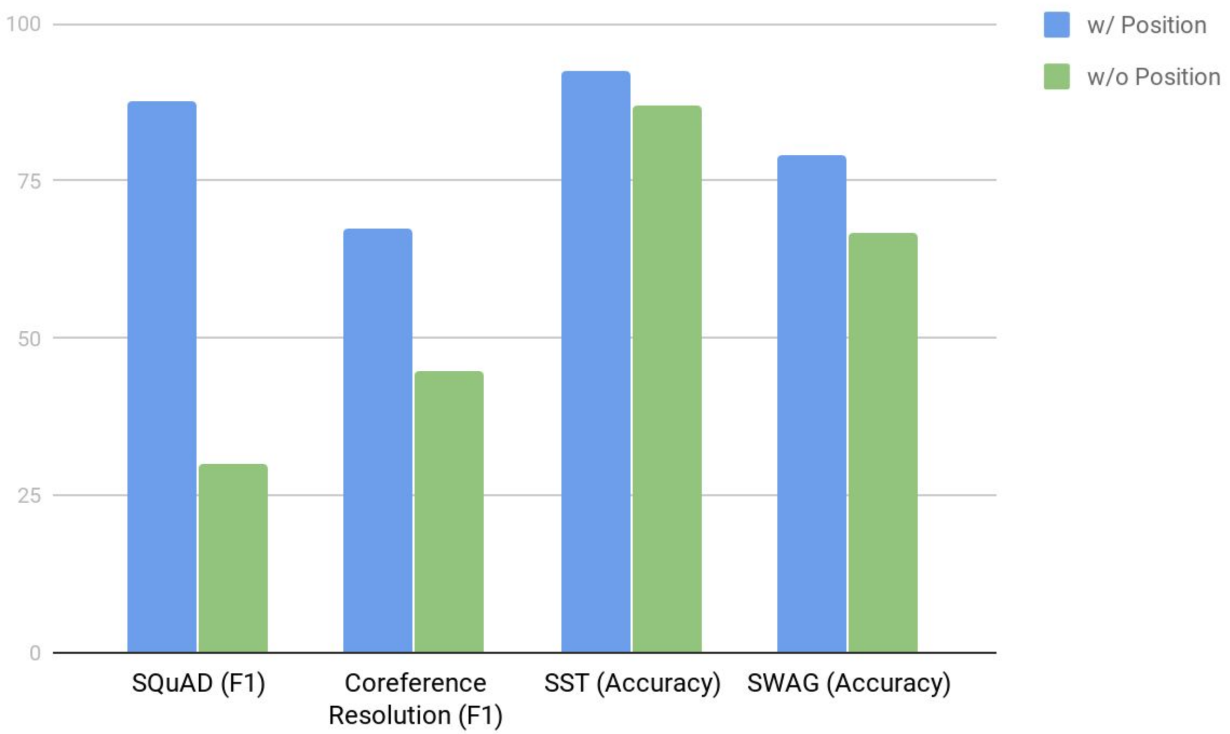
Results on removing position embeddings in BERT

Task Category	Task	with Pos	w/o Pos	Abs Diff	% Diff
Span Extraction	SQuAD (F1)	87.5	29.9	57.6	65.8
Input Tagging	Coreference Resolution (F1)	67.4	44.6	22.8	33.8
Sentence Decoding	CNN/Daily mail (Abstractive summarization)	0.191	0.109	0.08	42.9
Sentence Classification	CNN/Daily mail (Extractive summarization)	0.193	0.119	0.07	38.3
Classification	SWAG (Accuracy)	79.1	66.7	12.4	15.7
Classification	SST (Accuracy)	92.4	87.0	5.4	5.8
Classification	MNLI	80.4	76.9	3.5	4.4
Classification	MNLI-MM	81.0	76.8	4.2	5.2
Classification	RTE	65.0	58.8	6.2	9.5
Classification	QNLI	87.5	83.6	3.9	4.5

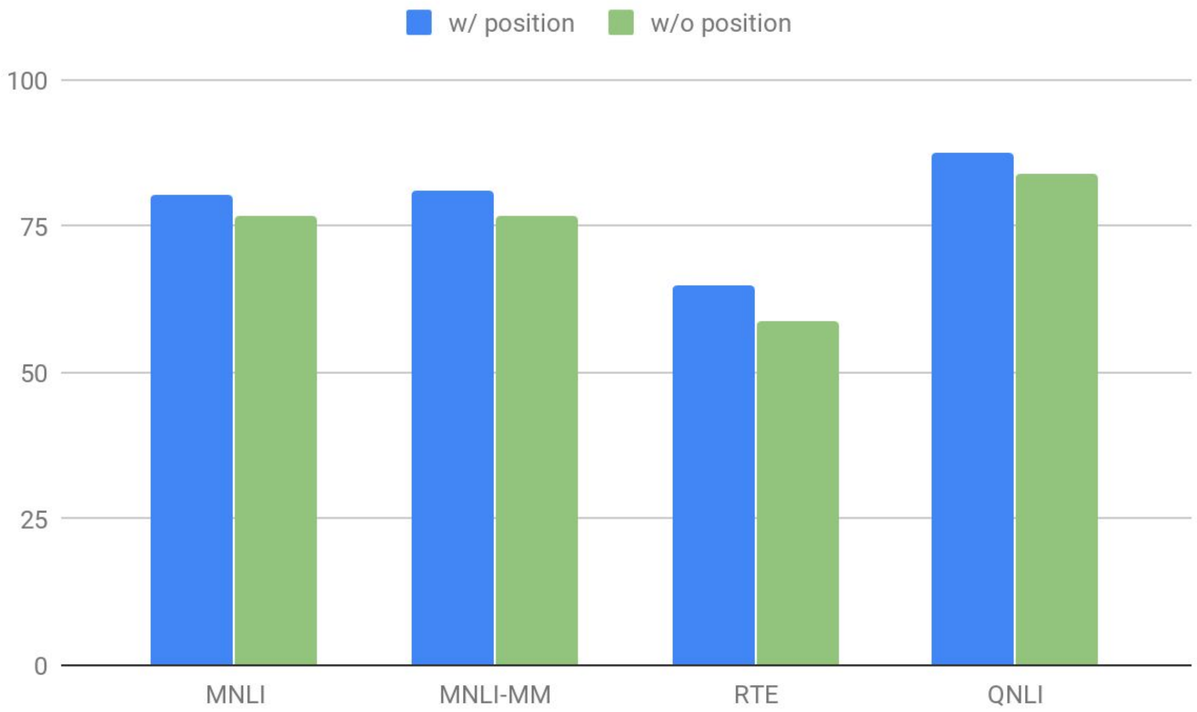
Summarization Results



Question Answering/Text Classification Results



Natural Language Inference



Observations

- Deeper layers capture less position information than earlier ones in BERT
- Position embeddings matter less for classification tasks
  - But are important for sequence-based tasks (sequence tagging, span prediction, etc.)

Observations

- Deeper layers capture less position information than earlier ones in BERT
- Position embeddings matter less for classification tasks
  - But are important for sequence-based tasks (sequence tagging, span prediction, etc.)

Next Steps...

- Finetune on downstream tasks with other perturbed training schemes
- Run position DC on finetuned models to see how they capture position
- Analysis of model errors from missing positional information